

*Scalable Deep Learning of Generative Models for
Molecular Design Optimization*

Jim Brase, LLNL
Sam Jacobs, LLNL
Brian Van Essen, LLNL

ATOM is an open and growing public-private partnership for accelerating drug discovery

Leadership

J. Baldoni – SVP GSK
(ret)

J. Brase – LLNL

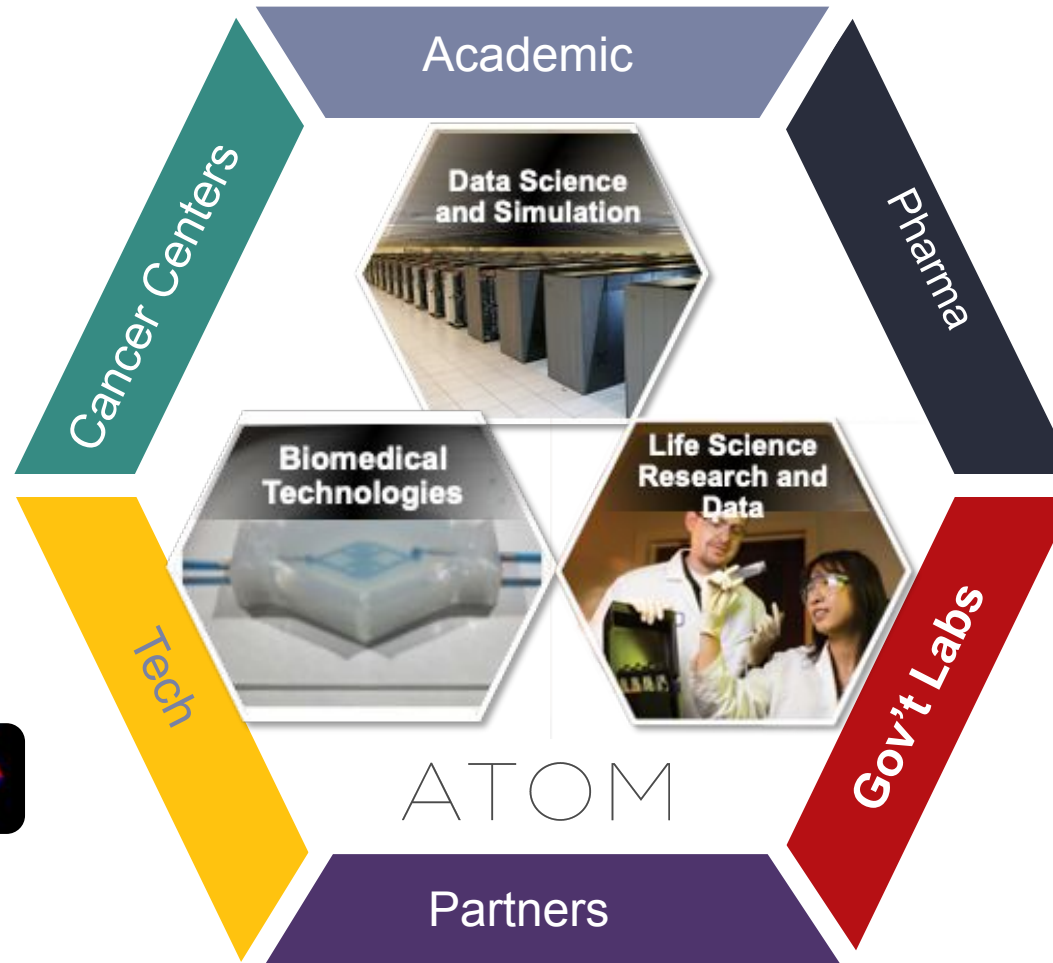
E. Stahlberg – FNL/NCI

M. Head – ORNL,

GSK (ret)
Lawrence Livermore
National Laboratory

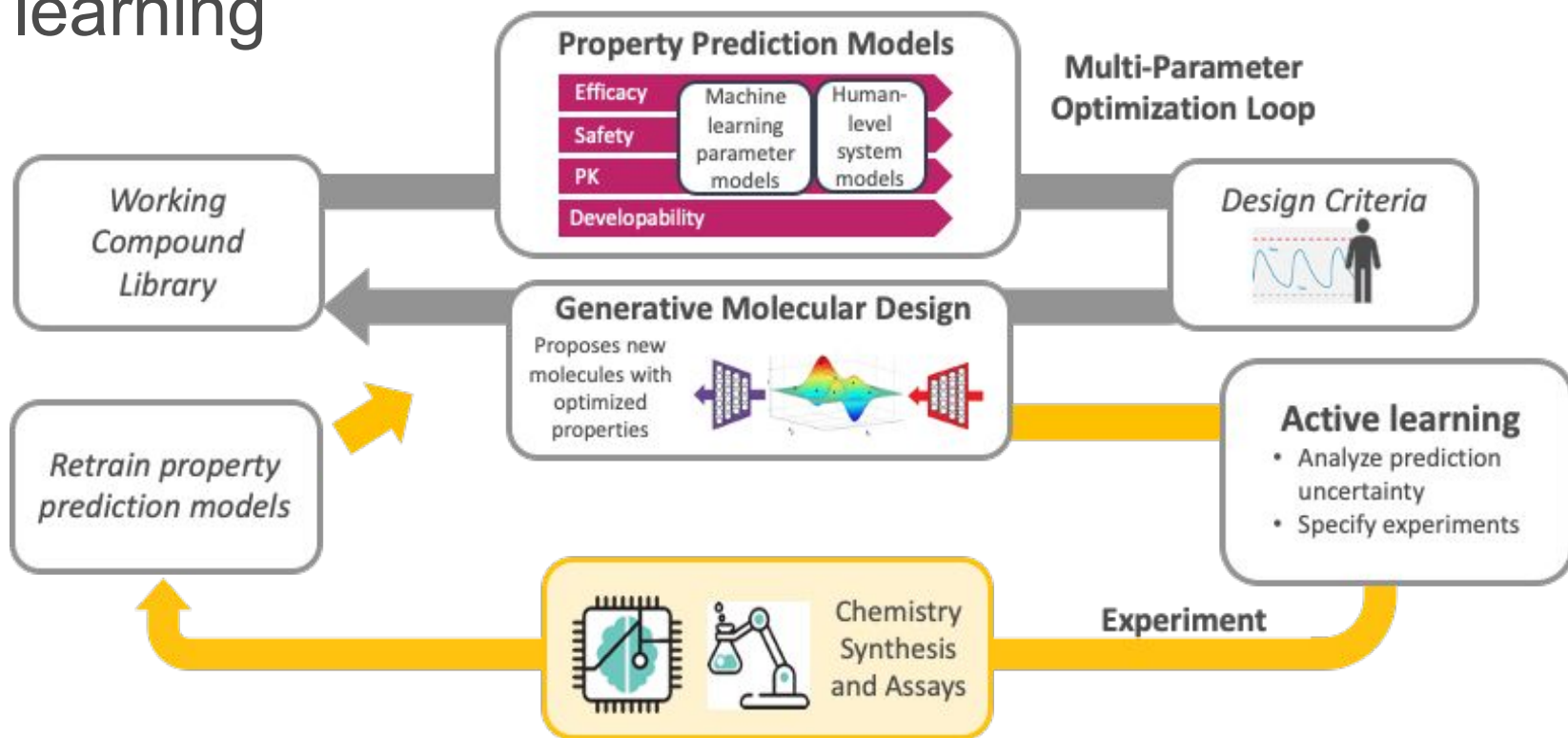


Frederick National Laboratory
for Cancer Research



ATOM

The goal of ATOM is to establish an open framework for generative molecular design with human-level predictive models and active learning



R&D Program Components

1. Establish open, curated data sets ready for modeling
 - Covering safety, PK, and efficacy for multiple targets
 - Partnerships to grow the data
2. Tools and frameworks for predictive modeling R&D
 - AMPL model training pipeline released
 - Extending to multi-scale human system models
3. Develop an open generative molecular design platform
 - High-performance multiparameter optimization (gray loop) in place
 - Demonstration and initial validation on AURK cancer target
 - Active learning loop (yellow) in progress
 - Pilots projects on COVID-19 set with partners

Status Summary

- Shared collaboration space at Mission Bay, SF
- Starting Year 4 of 5, ~20 FTEs engaged on R&D team
- New 501c3 structure with multiple pharma partners starting up in Jan 2021

AMPL has been released open source

Fork me on GitHub

README.md

ATOM Modeling PipeLine (AMPL) for Drug Discovery

license mit


Created by the *Accelerating Therapeutics for Opportunities in Medicine (ATOM) Consortium*

ATOM

AMPL is an open-source, modular, extensible software pipeline for building and sharing models to advance in silico drug discovery.

The ATOM Modeling PipeLine (AMPL) extends the functionality of DeepChem and supports an array of machine learning and molecular featurization tools. AMPL is an end-to-end data-driven modeling pipeline to generate machine learning models that can predict key safety and pharmacokinetic-relevant parameters. AMPL has been benchmarked on a large collection of pharmaceutical datasets covering a wide range of parameters.

A pre-print of a manuscript describing this project is available through [ArXiv](#). readthedocs are available as well [here](#).

 Cornell University

We gratefully acknowledge support from the Simons Foundation and member institutions.

arXiv.org > q-bio > arXiv:1911.05211

Search... All fields Search

Help | Advanced Search

Quantitative Biology > Quantitative Methods

AMPL: A Data-Driven Modeling Pipeline for Drug Discovery

Amanda J. Minnich, Kevin McLoughlin, Margaret Tse, Jason Deng, Andrew Weber, Neha Murad, Benjamin D. Madej, Bharath Ramsundar, Tom Rush, Stacie Calad-Thomson, Jim Brase, Jonathan E. Allen

(Submitted on 13 Nov 2019 (v1), last revised 14 Nov 2019 (this version, v2))

One of the key requirements for incorporating machine learning into the drug discovery process is complete reproducibility and traceability of the model building and evaluation process. With this in mind, we have developed an end-to-end modular and extensible software pipeline for building and sharing machine learning models that predict key pharmaceutical parameters. The ATOM Modeling PipeLine, or AMPL, extends the functionality of the open source library DeepChem and supports an array of machine learning and molecular featurization tools. We have benchmarked AMPL on a large collection of pharmaceutical datasets covering a wide range of parameters. As a result of these comprehensive experiments, we have found that physicochemical descriptors and deep learning based graph representations significantly outperform traditional

Download:

- PDF
- Other formats (license)





Current browse context:
q-bio.QM
< prev | next >
new | recent | 1911

Change to browse by:
cs
cs.LG
q-bio
stat
stat.ML

References & Citations

- NASA ADS

Export citation
Google Scholar

Bookmark
   

<https://arxiv.org/abs/1911.05211>

<https://github.com/ATOMconsortium/AMPL>

AMPL is the basis for ATOM student engagement programs

- **ATOM summer internships:**
 - Six students from Butler University PharmD and UC Davis
 - Planning expansion for Summer 2021
- **Purdue University Data Mine Program:**
 - Support a data science team (~10 students)
 - Focusing on data analysis and machine learning applications with AMPL
- **Six trainees among ATOM member labs**

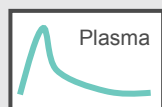
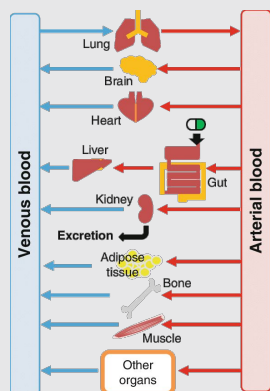
Summer 2020 Intern Projects

- *Data Curation for a Mitochondrial Membrane Potential Model*
- *Public Datasets within AMPL*
- *Visualize Data: Creating Interactive Plots to Improve Exploratory Data Analysis*
- *Working with Open Data Sources: PK DB, Lombardo Dataset, and AstraZeneca*
- *Featurization and Analysis of COVID-19 data*
- *Explainable 3D-CNN Models for Protein-Ligand Binding*

Predicting Volume of Distribution (VDss) in Humans

Performance of *in silico* Methods for a Large Set of Structurally Diverse Clinical Compounds

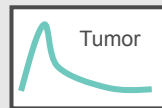
Pharmacokinetics (PK) Platform



- Developed physiologically-based pharmacokinetics (PBPK) model for human PK prediction



- Applied to human VDss predictions



Experimental Datasets

Datasets

Plasma protein binding

Blood plasma partitioning

LogD

Adipocyte partitioning

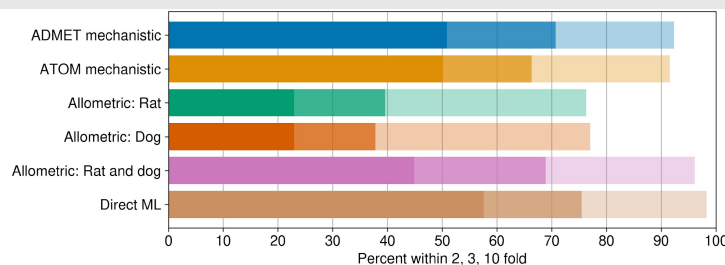
Myocyte partitioning

- Obach Lombardo (DMD 2018) compounds with human PK measurements
- ATOM collected *in vitro* PK data for 250 compounds
- Largest publicly-available *in vitro* and *in vivo* PK dataset

VDss Prediction Approaches

Input	Methods
2D molecular structures	1. Mechanistic models for tissue partitioning with predicted PK properties □ human VDss
	2. Mechanistic models for tissue partitioning using experimental PK properties □ human VDss
	3. Allometric scaling of predicted animal VDss
	4. Direct machine learning prediction of human VDss

In silico VDss Prediction Evaluation

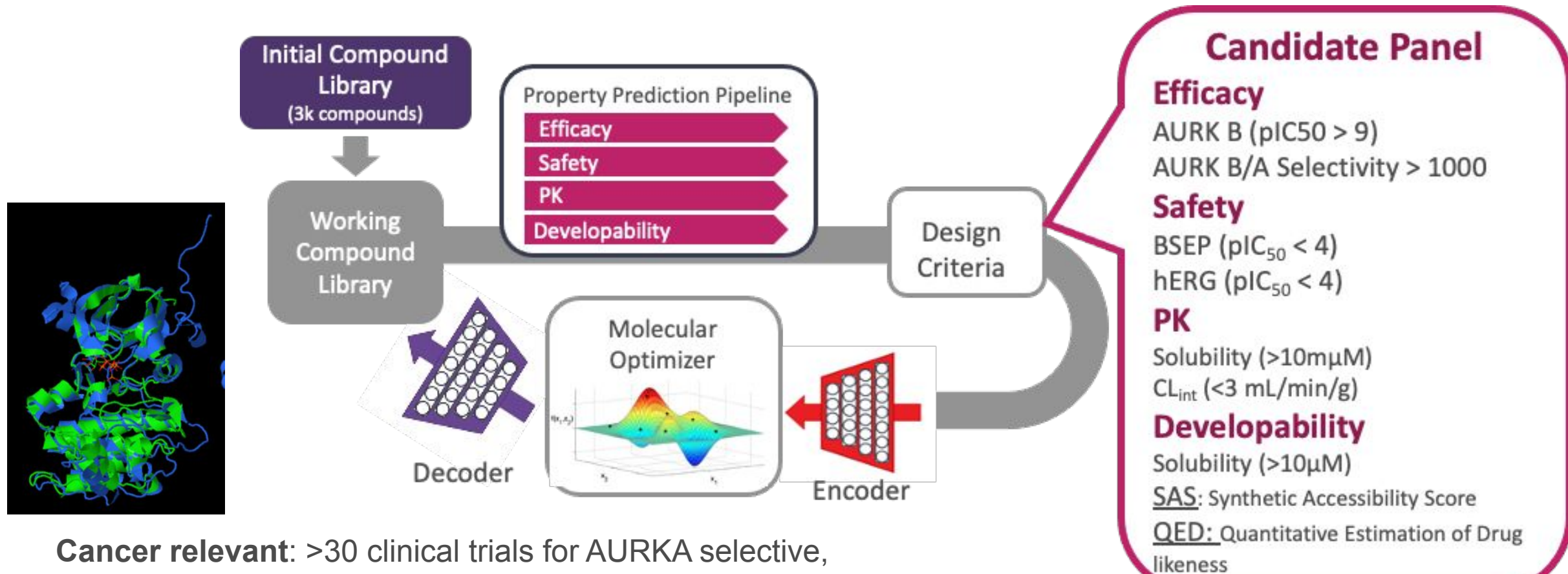


- Comparison to experimental test set
- For limited data, mechanistic models for tissue partitioning were most effective
- With enough animal and human VDss data, direct machine learning models were able to predict human VDss

ATOM Generative Molecular Design loop

Proof-of-Concept

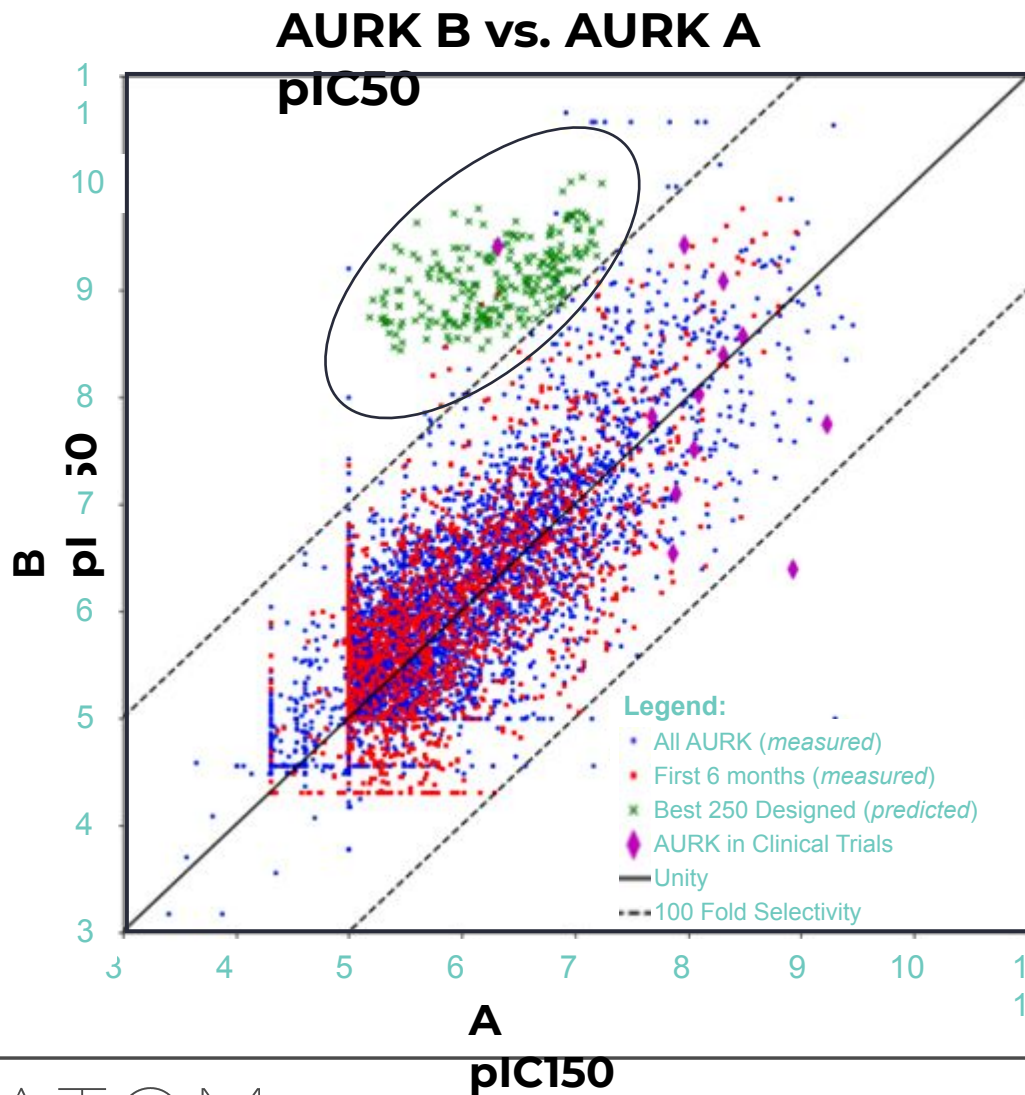
Generative molecular design of AURK B inhibitors



Cancer relevant: >30 clinical trials for AURKA selective, AURKB selective, and AURKA/B dual inhibitors

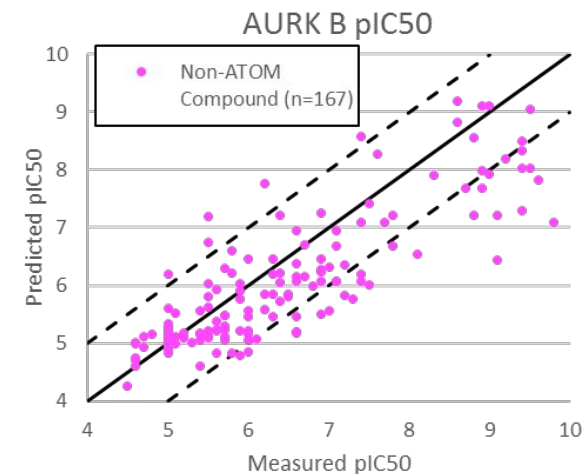
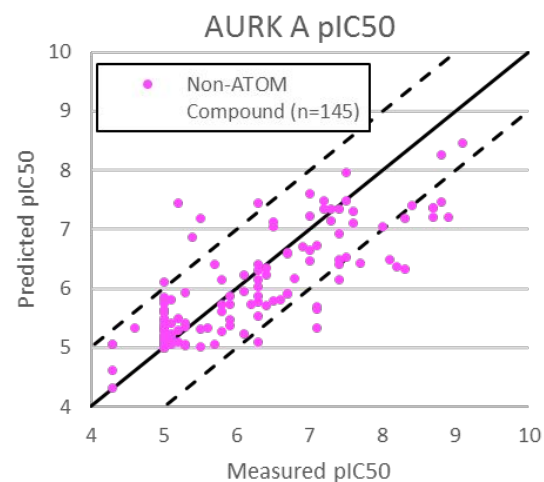
Data available at ATOM: Potency data on ~24k compound available for AURK B and/or AURK A

~200 Compounds with high potency, selectivity,
and other favorable properties



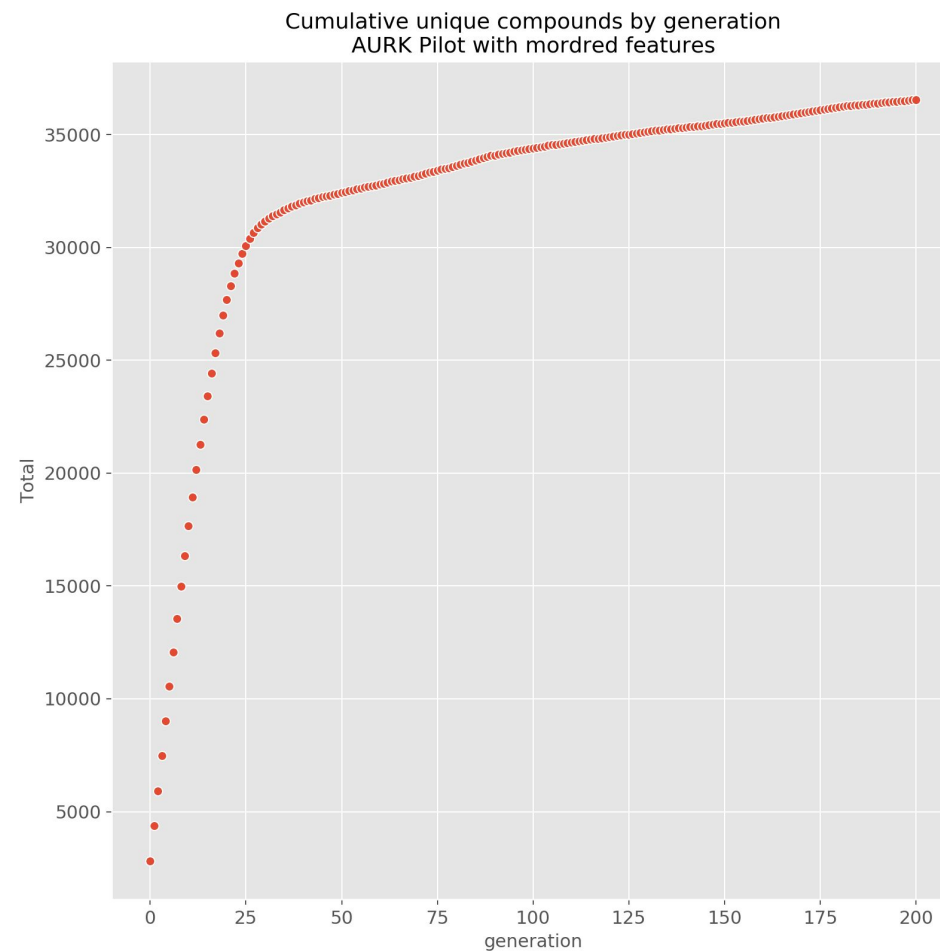
Efficacy of generated compounds **not in**
ATOM database is well predicted

R^2 : AURK A : **0.68** AURK B: **0.75**



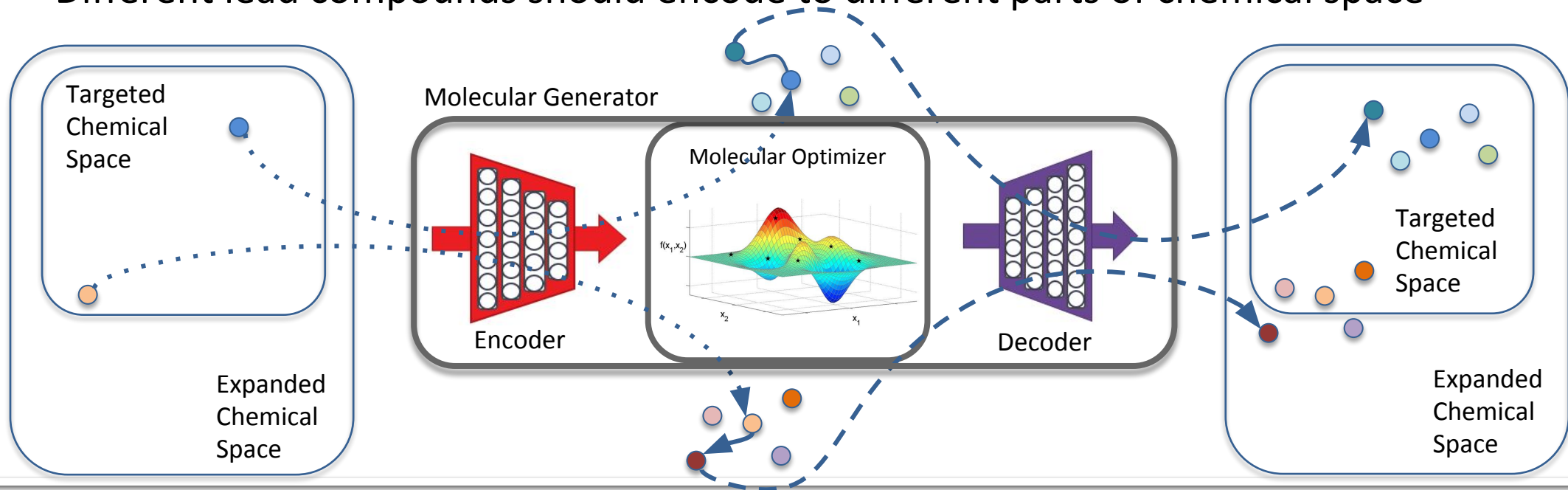
The generative model is creating a limited range of new molecules

- Number of novel generated compounds falls off after 25-40 generations
 - 600,000 evaluated, but only 36,553 unique
 - Few top-ranking molecules discovered after 40 generations.
 - Many in top 500 share common scaffold.
- Reasons:
 - Greedy genetic algorithm converges on narrow region of chemical space. Need to adjust mutation rate, other parameters.
 - VAE trained on project-specific compound set can only generate compounds with same “vocabulary”. Training on more diverse set will alleviate this limitation.

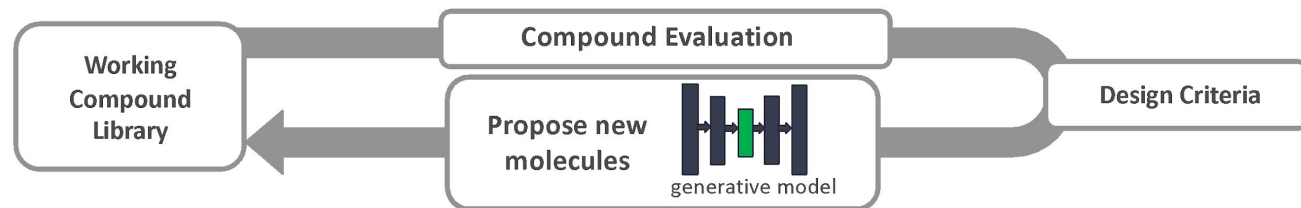


Training molecular generator with more compounds should lead to an increase in chemical diversity of proposed solutions

- Project a lead compound into the molecular generator's latent space
- Optimize compound in latent space through guided search and small perturbations
- Project new latent vector back into chemical space to create new compound
- Different lead compounds should encode to different parts of chemical space



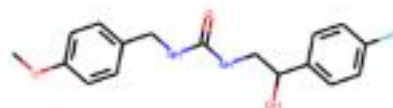
Our current encoders and decoders are trained on a limited data set



- State of the art Junction Tree Variational Autoencoders are slow to train
 - Direct realization of molecular graphs using tree decomposition and graph message passing network
 - 24 hours on 1M chemical compounds using community implementation in PyTorch

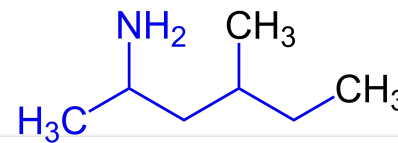
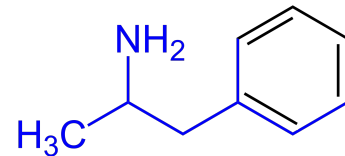
- Identify scalable neural network architecture
 - Explore new generations of character-based sequence models

- Encode the compound as SMILE string
 - Simplified molecular input line entry system



CN1C=NC2=C1C(=O)N(C(=O)N2C)C

- Perturb the chemical's latent representation
- Decode a new chemical
 - Generational auto-encoder architecture
- Evaluate chemical similarity to original compound
 - Use Tanimoto similarity / distance metric



Amphetamine and Methylhexanamine similarity.

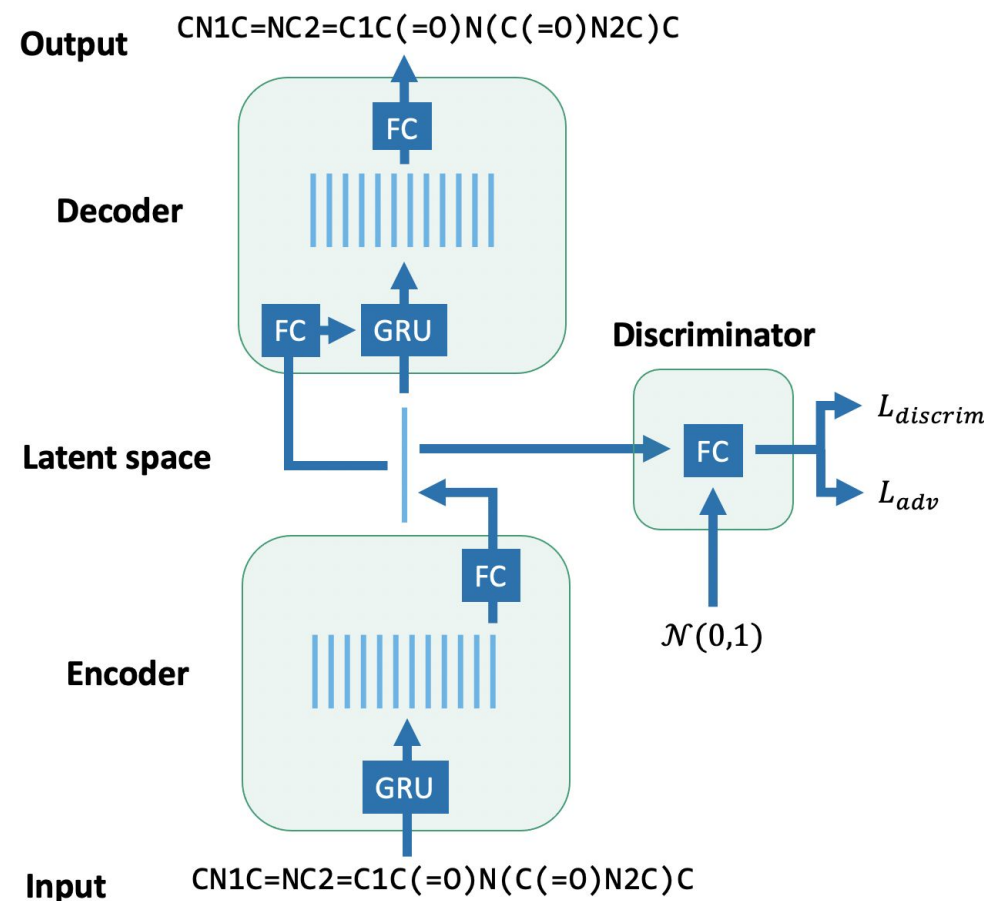
By Jü - Own work, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=31281660>

Expanding the diversity of molecular generation – scaling up the training data

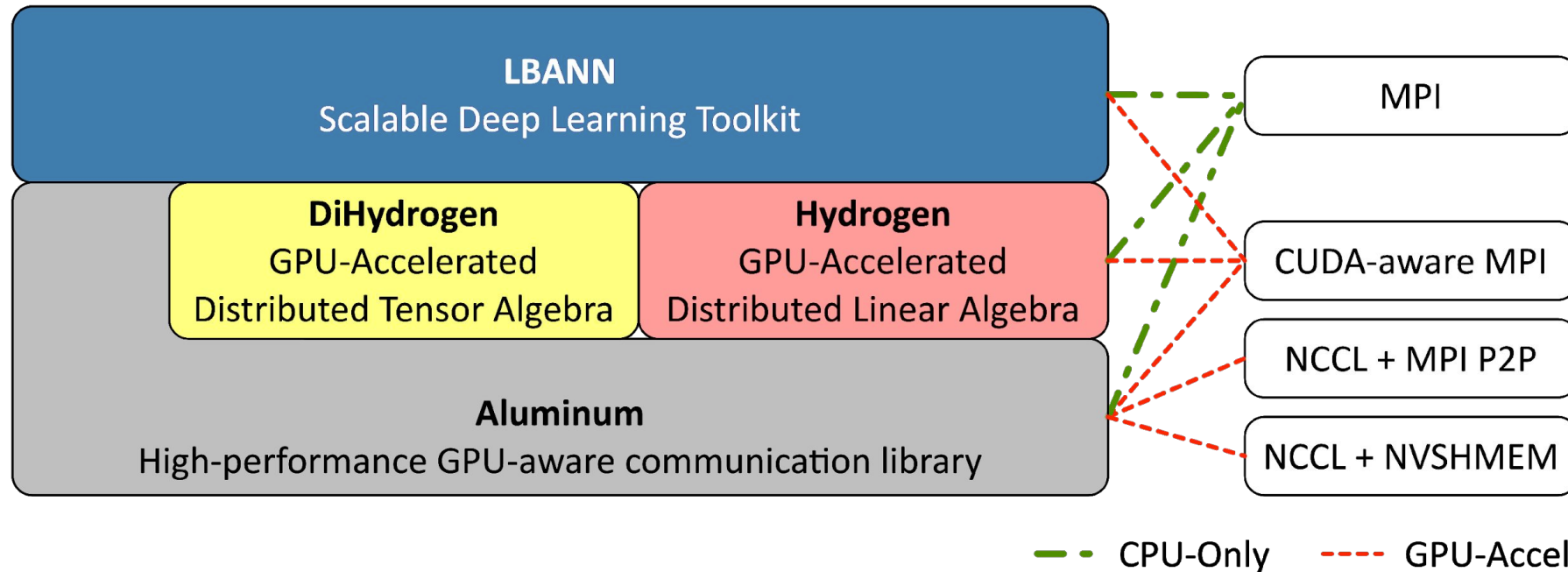
- In this work we will scale up to a training data set of 1.613B compounds
- Enamine REAL database (synthesizable drug-like compounds)
 - 1.36B chemicals - downloaded in the first quarter of 2020
- Enamine historical database
 - 252M non-overlapping historical Enamine compounds downloaded in the first quarter of 2018
- Targeted chemical compound database (Mpro_inhib)
 - 1 million additional purchasable compounds screened for SARS-CoV-2 main protease (Mpro) inhibition activity
- Our test set is composed of a held-out set of 2M Enamine and 10K Mpro_inhib compounds

Propose new character-Wasserstein Autoencoder (cWAE) that tackles issue of direct reconstruction required for lead optimization

- cWAE is a class of autoencoder that addresses problems with cAE and cVAE
 - character-based WAE version of original WAE [Tolstikhin et.al, 2017]
- Its basic structure is similar to cVAE but with a different additional regularized penalty terms
 - Additional term is implemented as a discriminator network D in latent space Z that differentiate between sample drawn from a Gaussian prior $\mathcal{N}(0,1)$ and samples drawn from latent space (Q_z)
- Wasserstein term **ensure better reconstruction** and **variational sampling** from latent space
- Variational latent space allows both guided and novel compound generation
- These characteristics enable small molecule design guided by lead optimization compound



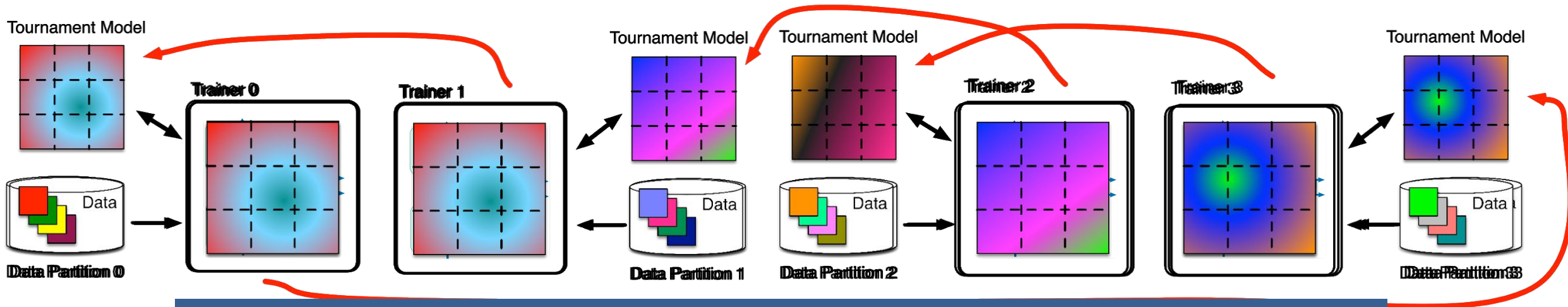
Training at Scale with 1.613B compounds and 16,640 GPUs on Sierra: Scalable Deep Learning Software Stack



- Optimized distributed memory algorithms
- Pythonic “PyTorch-based” model description
- Support for model exchange with PyTorch
- C++ / MPI + OpenMP / CUDA / cuDNN / NCCL
- Open-sourced on github.com
 - <https://github.com/LLNL/lbann>
 - <https://github.com/LLNL/Elemental>
 - <https://github.com/LLNL/DiHydrogen>
 - <https://github.com/LLNL/Aluminum>

The tournament method creates a single model instance that is trained on a massive data set [Jacobs et al. 2017, 2019]

- Entire supercomputer used to accelerate training
- Multiple trainers with independent, partitioned data sets
- Periodically exchange model with random peer
 - After one or more epochs of training
- Run local tournament to select current or exchanged model
- Continue training using winning model
- **Scalability** is maintained through parallelism
 - Within trainers: collective communication
 - Between trainers: point-to-point communication
- Benefits
 - Scalable peer-to-peer communication
 - Use parallel resources to reduce total time to train

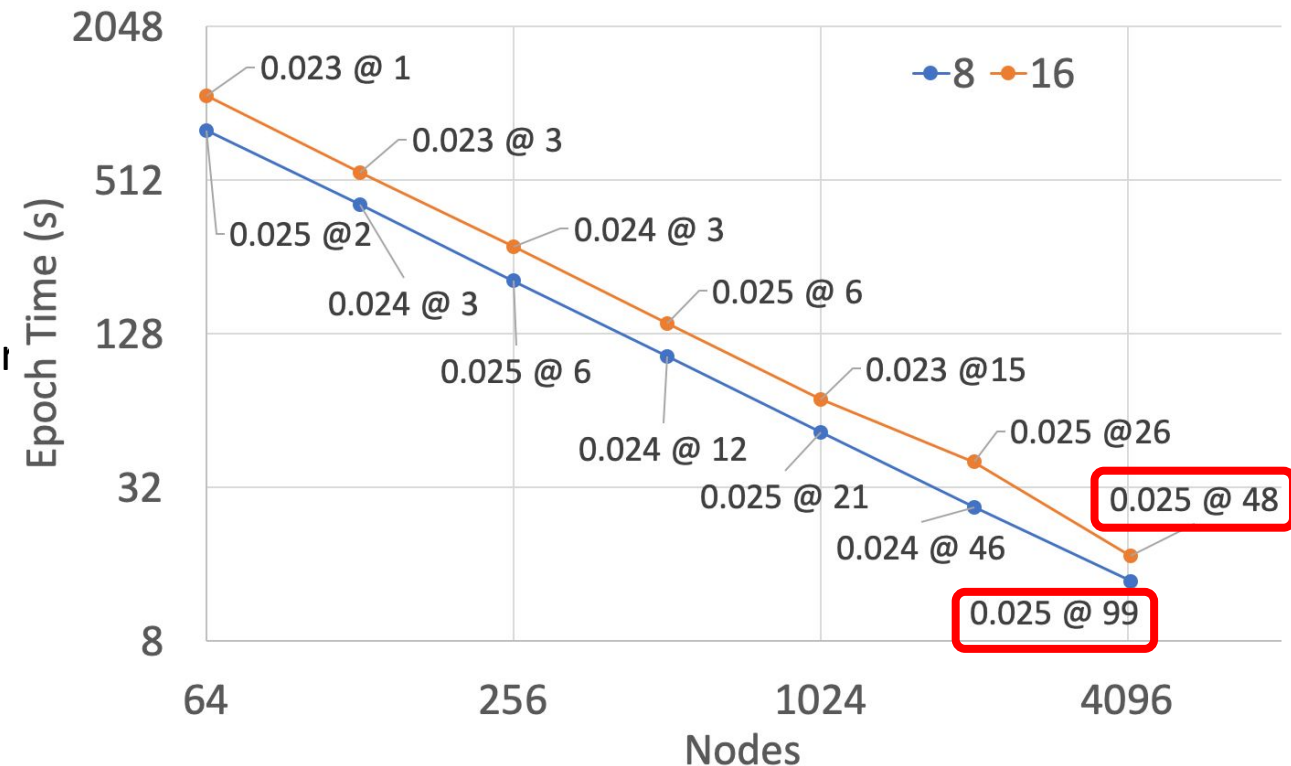


Scale up to largest HPC systems to train on the largest data sets

LBANN enabled Training of Models at Scales Previously Unobtainable

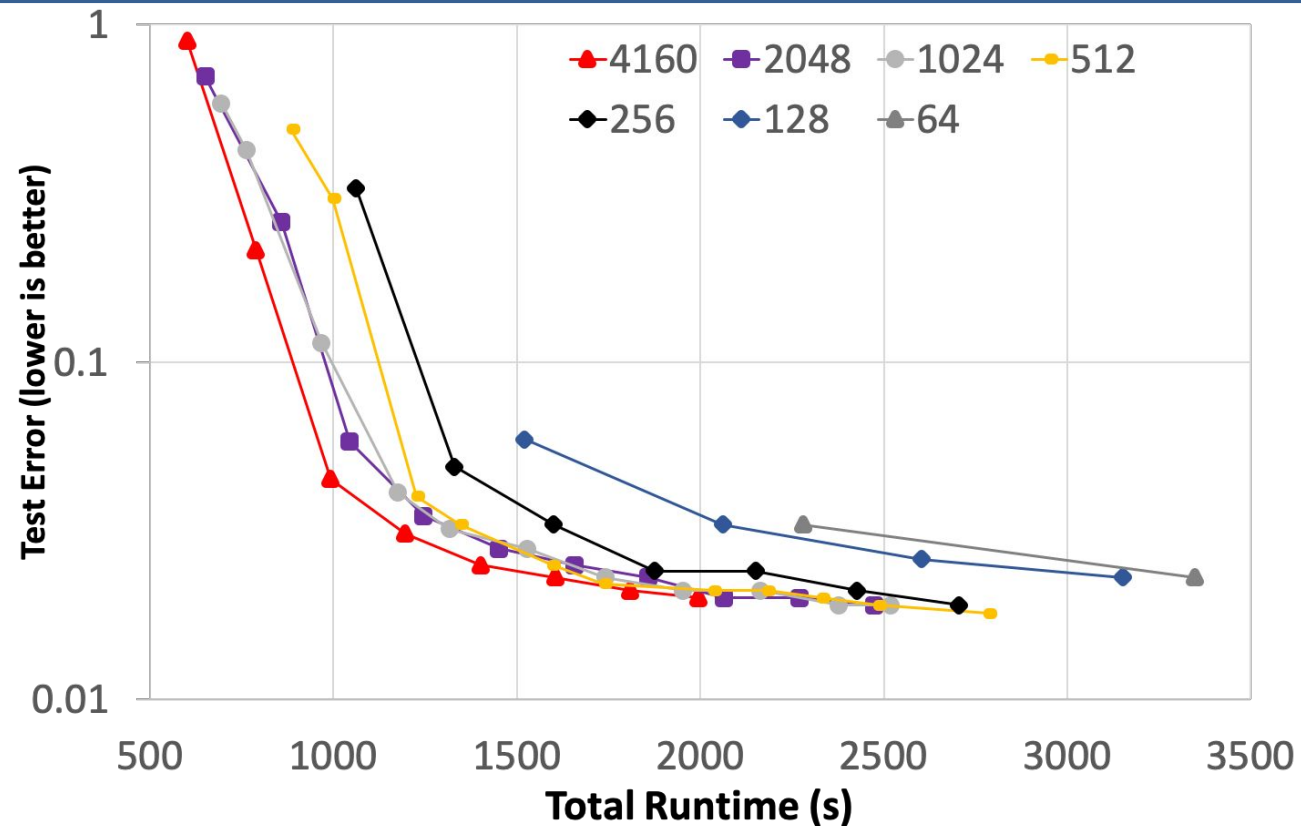
- Optimized data ingestion scaled training to 1.613B compound data set on up to 4160 nodes
- Reconstruction error of ~ 0.025 was good enough
 - Better than prior state of the art
 - Sufficient Tanimoto distance for domain scientists
- Asynchronous LTFB algorithm enables scaling without loss of model quality
 - Non-overlapping partition of the data set provided linear performance gains with more trainers
- Training time with 8 GPUs per trainer was faster per epoch
 - Took more epochs to achieve similar accuracy
- Reducing # of trainers reduced # of iterations required to meet fixed reconstruction error
 - 16 GPUs per trainer required fewer epochs

Test Error	% Valid	Recon. Rate	Avg. Tanimoto Distance
0.006	92.70	80.14	0.080
0.013	91.28	89.32	0.093
0.024	77.17	70.68	0.244
0.058	54.66	45.3	0.484



LBANN+LTFB enables improved time to solution with additional compute resources

- Increasing trainers reduces time required to train to fixed accuracy with 1.613B samples
 - 1.79× speedup at 256 nodes vs 64 nodes
 - Parallel efficiency of 44.65%
- Changes the time-to-insight from a compute-limited issue to a human-limited one
 - 32 minutes @ 256 nodes to 0.025 recon. error
 - 23 minutes @ 4160 nodes to 0.025 recon. error
- Running at 4160 nodes achieves 17.1% peak efficiency of FP16/acc FP32
 - 18% speedup for 4160 nodes vs 2048 nodes
 - 2.39× speedup with 4160 nodes vs 64 nodes
 - Parallel efficiency of 3.68%
- The ability to operate at this scale unlocks a new frontier for NN architecture design



GPUs/trainer	Trainers	Epoch time	PFLOPS
16	1040	17.2 s	253.3
8	2080	13.7 s	318.0

Table 6. Peak performance training with 4,160 nodes on Sierra.

cWAE Outperforms State-of-the-Art Junction-Tree-VAE for molecular reconstruction

- Improved Tanimoto distance ensures that reconstructed compounds have chemical similarity to original compound
- cWAE is faster to train and use for inference than JT-VAE
- Next steps are to integrate cWAE into ATOM design loop

Model	Train Size	Test Size	Percent Valid	Compound Reconstruction Rate	Average Tanimoto Distance
JT-VAE (state of the art)	1M Mpro	10K Mpro	100	1.63	0.553
cWAE	1M Mpro	10K Mpro	85.41	83.27	0.146
cWAE	1613M Combo	10K Mpro	42.45	33.03	0.601
cWAE	1613M Combo	10K Combo	92.70	90.14	0.080

Table 2. Summary of model accuracy. Metrics include percent of valid decoded SMILES strings (Percent Valid), Compound Reconstruction Rate and Average Tanimoto Distance between the encoded and decoded molecule (Average Tanimoto Distance, lower is better).

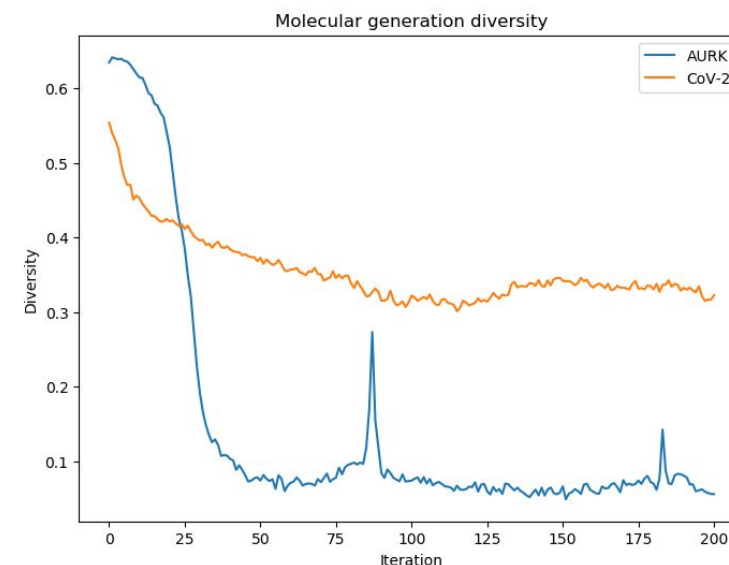
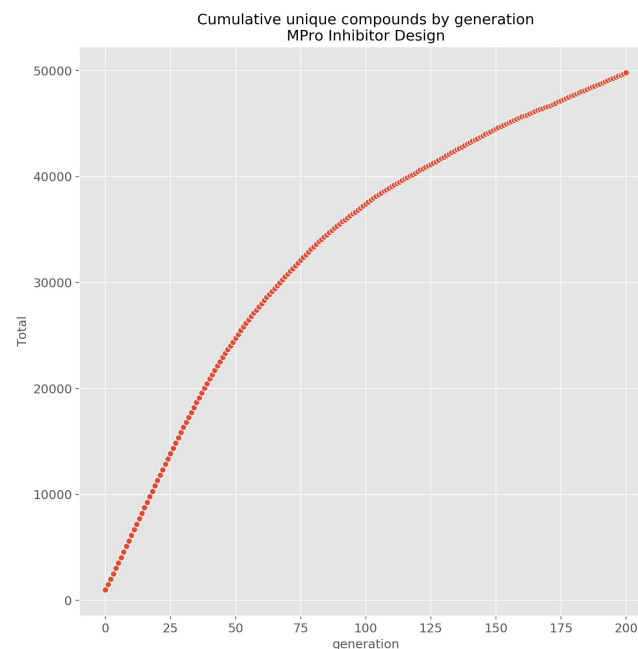
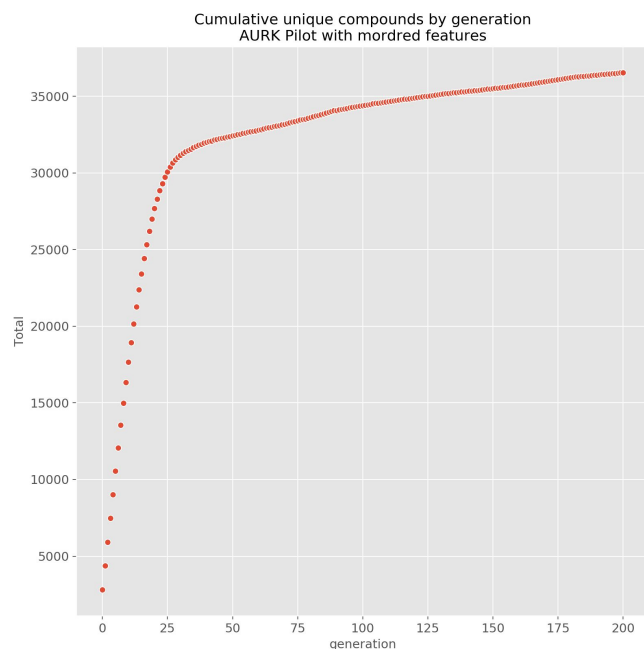
Lessons Learned: Deep learning at scale exposes unforeseen challenges -- Power swings

- What happens when the power company calls you to see what you just did?
- Deep learning at scale has center-wide impact → half-precision TensorCores lead to dramatic power swings:
 - Periodic 2-3 MW swings caused concern from power company
 - Asynchronous learning algorithm minimized center-wide power swings
 - Reduced power swing from >200KW per-row to <120KM (each row is 20 racks with 360 nodes)



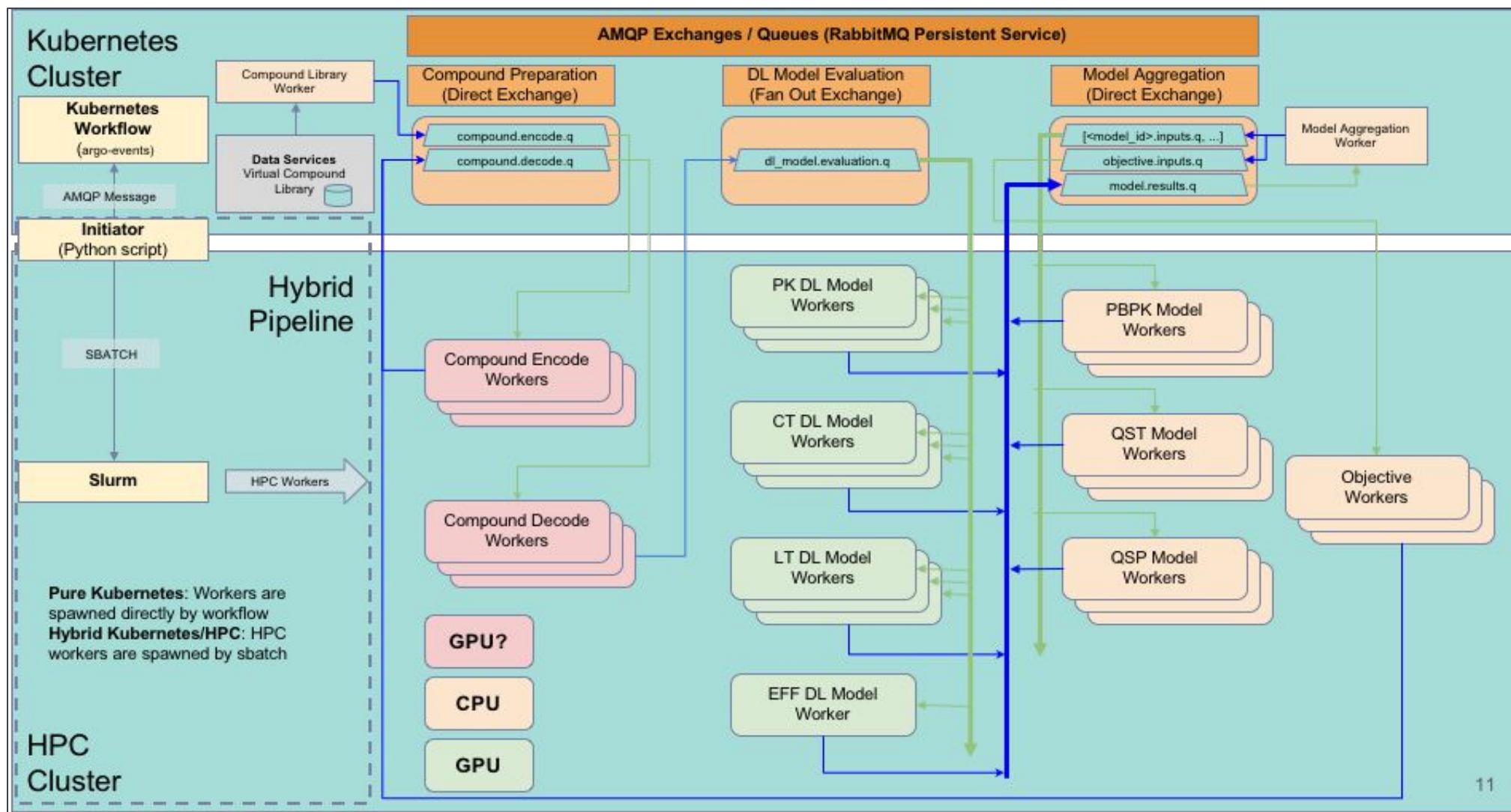
ATOM

Comparing unique compound generation and diversity from AURK pilot to current SAR-CoV-2 pilot



cWAE autoencoder provides improved diversity over trained chemical space

The ATOM generative molecular design loop runs in a hybrid cloud – HPC workflow environment



LBANN: Livermore Big Artificial Neural Network Toolkit

- Deep Neural Network training / classification
 - Optimize for strong & weak scaling
 - Train large networks quickly
 - Enable training on data samples or data sets too large for other frameworks
 - Billion sample data sets
 - Optimized distributed memory algorithms
 - Multi-level parallelism (model / data / ensemble)**

Nomenclature:

- Trainer:
 - Unique set of HPC resources
 - Contains one or more neural network models
 - Independent set of data
 - Implements model and data parallelism

